(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 March 2002 (14.03.2002)

PCT

(10) International Publication Number
WO 02/21763 A1

(51) International Patent Classification[7]: H04L 9/00

(21) International Application Number: PCT/US01/28308

(22) International Filing Date:
10 September 2001 (10.09.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/231,334 8 September 2000 (08.09.2000) US

(71) Applicant: MAINSTAY ENTERPRISES, INC. [US/US]; 209 West Street, Suite 204, Annapolis, MD 21401 (US).

(72) Inventors: ALUZZO, Gaspare; 1698 Epping Farms Lane, Annapolis, MD 21401 (US). BIGNO, William; 15292 Grist Mill Terrace, Woodbridge, VA 22191 (US).

(74) Agents: GARRETT, Arthur, S. et al.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US).

(81) Designated States *(national)*: CA, MX.

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 02/21763 A1

(54) Title: SYSTEM AND METHOD FOR PROTECTING INFORMATION STORED ON A COMPUTER

(57) Abstract: An embodiment of the invention is a method for protecting information processed, transmitted, or stored on a computer having a bootup operation, an operating system, at least one input device, and at least one output device. The computer checks for a first authentication during the bootup operation of the computer, the first authentication being associated with an external device connected to the computer. At least one input device of the computer is locked if no first authentication is present during the bootup operation of the computer. The computer may check for a second authentication when the operating system is operational and lock the at least one input device if no second authentication is present while the operating system is operational. Checking for an authenticated token as an authentication may comprise querying at a Smart Card reader for an authorized Smart Card.

SYSTEM AND METHOD FOR PROTECTING INFORMATION STORED ON A COMPUTER

BACKGROUND OF THE INVENTION

Related Applications

[001] This application claims the benefit of priority from Provisional U.S. Patent Application Serial No. 60/231,334, entitled "SYSTEM AND METHOD OF PROTECTING INFORMATION STORED ON A COMPUTER," filed on September 8, 2000, by the same inventors, which is expressly incorporated herein by reference.

[002] Additionally, this application is related to commonly owned U.S. Patent Application Serial No. _____, entitled "SYSTEMS AND METHODS FOR PROTECTING INFORMATION ON A COMPUTER BY INTEGRATING BUILDING SECURITY AND COMPUTER SECURITY FUNCTIONS," filed on the same date herewith by the same inventors.

Field of the Invention

[003] The present invention relates to computer security, and, more particularly, to the use of an authenticating device, such as a token, to secure a computer and its memory components from improper access.

Background and Material Information

[004] Because government and business entities are storing a wealth of sensitive information on computers, the protection of information has never been more important. Systems which safeguard data from espionage and theft have been developed to address this problem.

[005] On the level of personal computers, traditional systems often implement this protection via the use of passwords. Nonetheless, passwords for computer systems often fail their objective because they operate at the operating system (OS) level. An OS is a main control program for a computer that schedules tasks, manages storage, and handles communication with peripherals. The OS presents a basic user interface when no applications are open, and all applications must communicate with the operating system, thus allowing input from a user and output from application programs. A user desiring to bypass a password at the OS level often needs only to reboot the computer and/or operate the computer in a "Safe Mode" to avoid the OS level password protection. A Safe Mode is a special mode of the OS that loads with minimal driver support. The intended purpose of the Safe Mode is to help resolve boot problems, but its lack of security capability leaves open the possibility of unauthorized access to sensitive and/or proprietary information.

[006] Still other systems utilize a physical key to be used each time a computer is utilized. In these systems, the key is tied to the power supply of the computer so as to disable the power supply when the key is not present. Nonetheless, implementations of physical keys do not provide protection after a bootup sequence without requiring the computer to be powered down. "Bootup" refers to the process of loading the OS into the dynamic memory of the computer. For example, when a user leaves the user's computer terminal during the day while the computer is running, the user must either leave the computer running with the key present or power the computer down and remove the key. As such, it would simply be impracticable to turn the computer off and remove the key every time the user leaves the computer,

since a lengthy reboot operation would be required each time the user returned and replaced the key. More often than not, a computer user under this security regime would simply leave the computer running and would not remove the key during breaks, thus exposing the computer data to possible thievery.

[007] Moreover, traditional systems such as passwords or physical keys offer no protection against the physical removal and theft of a computer hard drive. When a thief takes a hard drive in this way, the thief often need only operate the hard drive as a "slave" to another computer to download and access all the information on the stolen hard drive.

[008] What is needed is a security system that allows for the protection of computer data or information upon bootup (*e.g.*, during the execution of the basic input/basic output (BIOS) instructions) *and* during the ongoing operation of the computer (*e.g.*, during the execution of the OS). Those skilled in the art will appreciate that BIOS instructions, which provide the initial processing instructions to the computer, are typically resident in the motherboard of a computer but can be stored elsewhere according to the particular computer architecture implementation.

[009] What is also needed is a system which provides effective security measures that allow for computer users to take breaks away from the computer without powering the computer down to activate the security upon leaving. Correspondingly, such security measures should not require a bootup operation to deactivate the security upon returning. Finally, this system should afford protection against the theft of computer data or

information even when a hard drive or other computer-readable memory is physically removed from the computer and manipulated by another computer.

SUMMARY OF THE INVENTION

[010] Methods, systems, and computer-readable media consistent with the present invention overcome these problems and provide enhanced security measures. In one aspect, a method consistent with an embodiment of the invention is disclosed for protecting information processed, transmitted, or stored on a computer having a bootup operation, an operating system, at least one input device, and at least one output device. The computer checks for a first authentication during the bootup operation of the computer, the first authentication being associated with an external device connected to the computer. At least one input device of the computer is locked if no first authentication is present during the bootup operation of the computer. The computer may check for a second authentication when the operating system is operational and lock the at least one input device if no second authentication is present while the operating system is operational.

[011] In another aspect of the present invention, a system consistent with an embodiment of the invention is described for protecting information in a computer having a memory and at least one input device. The system comprises at least one token, a reader for reading the at least one token, and a first software operational in the memory for disabling the at least one input device of the computer during a bootup operation if the at least one token is not read by the reader. The system may also comprise a second software operational in the memory for disabling the at least one input device of the

computer when an operating system is operational if the at least one token is not read by the reader.

[012] Moreover, the system may comprise a third software operational in the memory for disabling at least one output device of the computer when an operating system is operational if the at least one token is not read by the reader. Finally, the system may comprise a fourth software operational in the memory for placing the memory in an inaccessible state if the at least one token is not read by the reader when the computer attempts to access the memory.

[013] Additional advantages of the invention will be set forth in part in the description which follows, and in part will be obvious from the description, or may be learned by practice of the invention. The advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

[014] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[015] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and together with the description, serve to explain the principles of the invention.

[016] Reference will now be made in detail to the present embodiments of the invention, examples of which are illustrated in the

accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

[017] In the drawings:

[018] FIG. 1 is a diagram of an exemplary system environment in which to practice an embodiment of the present invention;

[019] FIG. 2 is a flowchart of an exemplary method in accordance with an embodiment of the present invention;

[020] FIG. 3 is a flowchart of an exemplary subroutine in accordance with an embodiment of the present invention;

[021] FIG. 4 is a flowchart of another exemplary subroutine in accordance with an embodiment of the present invention;

[022] FIG. 5 is a flowchart of yet another exemplary subroutine in accordance with an embodiment of the present invention;

[023] FIG. 6 is a flowchart of an exemplary method for utilizing public key infrastructure (PKI) in accordance with an embodiment of the present invention; and

[024] FIG. 7 illustrates a flowchart of an exemplary method for checking to determine whether a user is authenticated to access files in static memory consistent with an embodiment of the present invention.

DESCRIPTION OF THE EMBODIMENTS

[025] Embodiments of the present invention provide for a system which protects information stored within or as part of a computer by requiring an authentication in order for a user to access the information. The authentication is associated with an external device connected to the

computer. For example, the external device may comprise a token reader, such as a Smart Card reader, which reads tokens provided by one or more authorized users. Further, software is loaded into the memory of the computer which provides protection at both the Basic Input/Output System (BIOS) level and at the OS level. Coding resident in the memory of the computer provides another means of protection. This coding ensures that the memory, e.g., the hard drive, cannot be removed from the computer and used as a slave to another computer.

[026] Referring to FIG. 1, an exemplary system environment is illustrated in which to practice an embodiment of the present invention. In this exemplary embodiment, a computer 102 includes a memory 104, at least one input device 108, and at least one output device 110. Input device 108 may comprise, by way of example but not of limitation, a mouse, a keyboard, a microphone, a joystick, a trackball, a touch screen reader, a scanner, and the like. Output device 110 may comprise, by way of example but not of limitation, a printer, a plotter, a monitor, a speaker, and the like.

[027] Computer 102 is operatively connected to a token reader 114 by link 112. One skilled in the art will recognize that the connection may include a port or communication interface (not shown) comprising, among other things, switches. Nevertheless, the present invention contemplates any suitable implementation of a communication interface. A token 116 is to be used with reader 114 for providing authenticated access to computer 102. Token reader 114 may comprise, for example, a Smart Card reader, a radio frequency (RF) tag reader, a magnetic stripe reader, a retinal scanner, a fingerprint reader, a palmprint reader, and the like. Correspondingly, token

116 may comprise, for example, a Smart Card, an RF tag, a token having a magnetic stripe, or the retina, fingerprint, or palmprint from a user (not shown). Generally, token 116 stores (by design or merely by coincidence) the encoded authenticating information for the user.

[028] When token 116 is placed in contact with, in proximity of, or in operative engagement with token reader 114, token reader 114 verifies authenticating information encoded in or on token 116. This authenticating information is then sent to computer 102 via link 112 and a communication interface (not shown) of computer 102.

[029] In the exemplary embodiment, token reader 114 may secure and retain token 116 during the process of authenticating a token 116 described above. Furthermore, token reader 114 may secure and retain token 116 during the usage of computer 102 after token 116 is authenticated. In this way, when token 114 is removed from token reader 114, computer 102 may check for authenticity and disallow further use of the input device 108, the output device 110, and/or any other computer functionality. Token 116 may also be fixably attached to the user or the user's person, such that when the user leaves the computer terminal, token 116 is removed from token reader 114. In this way, authenticated access to the input device 108, the output device 110, and/or any other computer functionality will be given only when token 116 is present in token reader 114.

[030] Thus, software 118 operational in memory 104 may disable at least one input device 108 during a bootup operation if an authenticated token 116 is not present in reader 114. Alternately, software 118 operational in memory 104 may disable at least one output device 110 during a bootup

operation if authenticated Smart Card 116 is not present in reader 114. Moreover, at least one input device (such as input device 108) and/or at least one output device (such as output device 110) may be disabled when an authenticated Smart Card 116 is not present in reader 114 while the OS is operational in memory 104.

[031] In an exemplary embodiment utilizing public key infrastructure (PKI) for encryption and/or decryption of data, computer 102 sends authenticating information to a third party certification authority 122 via link 120. Third party certification authority 122 may then send an electronic permission indicator back to computer 102 via link 120. Third party certification authority 122 may comprise any a trusted third party responsible for issuing authentications, such as Entrust, VeriSign, Baltimore Technologies, and RSA Security, for example. When computer 102 receives the electronic permission indicator, computer 102 is then authorized to either encrypt or decrypt data.

[032] Turning to FIG. 2, an exemplary method in accordance with an embodiment of the present invention will be illustrated. In step 200, a computer is powered on or otherwise booted. BIOS instructions are read from read-only memory (ROM) to random access memory (RAM) in step 202. Along with these BIOS instructions, additional program instructions are read into RAM relating to token reader 114 (step 204). These extra instructions may be read into RAM immediately after BIOS instructions are read into RAM. Thus, instructions for both the BIOS and Smart Card system are executed substantially coincidentally, as shown in step 206.

[033] When these instructions are executed, the computer performs a check to see if a token reader is present (step 208). A determination is made in step 210 whether a token reader is present. If the reader is present, the method advances to the subroutine of FIG. 3 described below (step 212). If the reader is not present, the method advances to the subroutine of FIG. 4 described below (step 214).

[034] An exemplary subroutine in accordance with an embodiment of the present invention will now be described with reference to FIG. 3. This exemplary subroutine is typically performed only if a token reader is present in step 212 of FIG. 2. At step 300, a check is performed to determine if an authenticated token is operatively engaged at token reader 114. In step 302, a determination is made whether an authenticated token is present at the token reader 114. If an authenticated token is present, the subroutine advances to step 304. Normal bootup is continued in step 304 with full functionality for the at least one input device 108. If an authenticated token is not present, however, the subroutine advances to step 306, where a port or other type of communication interface corresponding to the at least one input device 108 is locked.

[035] The port may be locked via an instruction (*e.g.*, a binary instruction) implemented in software which may, for example, engage a switch. By engaging the switch, all signals coming from the at least one input device are interrupted, effectively locking the at least one input device. Step 306 serves to disable the at least one input device 108 so that a would-be thief (more generally referred to as a potential user) would be thwarted in using the input devices of the computer to circumvent security measures.

Thus, the computer is locked at the BIOS level (*i.e.*, before the OS is operational) because a user cannot manipulate the computer even when the BIOS is operational. In addition, in the case where an unauthenticated token has been presented at token reader 114, step 306 may further comprise detaining the unauthenticated token in the computer access token reader 114.

[036] In step 308, conventional OS files and startup files are loaded from ROM or from static memory to RAM, and these files are executed in a normal fashion. For example, in a Microsoft DOS operating system, CONFIG.SYS, AUTOEXEC.BAT, and other various OS files are typically processed at this step. Those skilled in the art will be familiar with the bootup process for the Microsoft DOS operating system as well as for other computer systems.

[037] The method advances to step 310, where a check is performed to determine if an authenticated token is operatively engaged at token reader 114. The purpose of this check is to determine whether a reader may have been connected by a would-be thief, only to be removed once the BIOS security checks have been surpassed. In contrast to the previous check, which operated at the BIOS level, this check provides a measure of protection at the OS level. In step 312, a determination is made whether the token reader is still present to the computer system. If a token reader is not present at this check, the method advances to step 316, where the at least one input device 108 is locked. In the exemplary embodiment, locking of the at least one input device 108 is accomplished in the same manner described above: A binary instruction engages a switch, thereby interrupting signals from the at least one input device.

[038] Step 316 effectively disallows the user from manipulating computer functions in any way. Further, in step 318, a default screen is displayed which overrides normal computer displays. This default screen may comprise, for example, a monochromatic screen display or a default message display (*e.g.*, "Computer Locked"). Optionally, the monitor may be powered down by the computer, revealing a blank screen. If a token reader is present at step 312, the method advances to the subroutine shown in FIG. 5, as described below (step 314).

[039] Now with reference to FIG. 4, an exemplary subroutine according to an embodiment of the present invention will be described. The exemplary subroutine of FIG. 4 corresponds to the situation where a token reader is not present at the first check. In step 400, ports corresponding to the at least one input device 108 are locked, thus preventing a computer user from manipulating computer function in any way. In step 402, conventional OS files and startup files are loaded from ROM or from static memory to RAM, and these files are executed in a normal fashion. The method advances to step 404, where a check is performed for the presence of the token reader at the token reader port. The purpose of this check is to determine whether a reader may have been connected by a would-be thief, only to be removed once the BIOS security checks have been surpassed.

[040] In step 406, a determination is made whether the token reader is still present to the computer system. If a token reader is not present at this check, the method advances to step 410, where the input devices are locked. Further, in step 412, a default screen is displayed which overrides normal

computer displays. If a token reader is present at step 406, the method advances to the subroutine shown in FIG. 5, as described below (step 408).

[041] FIG. 5 illustrates a flowchart of a subroutine in accordance with the present invention. In step 500, a check is performed to determine whether an authenticated token is operatively engaged at the token reader. In step 502, a determination is made whether an authenticated token is present at the token reader 114. If an authenticated token is not present at this check, the method advances to step 506, where the input devices are locked and step 508, where a default screen is displayed, the default screen typically overriding normal computer displays. If an authenticated token is present at step 502, normal computer function is available (step 504).

[042] Once a user has access to full computer functionality, the computer may perform an ongoing check that the authenticated token is still present. This is represented by the loop back from step 504 to step 500 in FIG. 5. In this way, when an authenticated user removes an authenticated token from the token reader 114, the ongoing check will cause the computer to immediately engage the locking feature of step 504 and the default screen of step 508.

[043] In practice, embodiments of the present invention allow a user to boot a computer even when an authenticated token is not present. In this situation, however, the input devices are locked during bootup, so as to thwart an unauthorized user's attempt to override the bootup and to place the computer in a "Safe Mode," for example. Therefore, the computer cannot be manipulated by an unauthorized user during the bootup sequence. Correspondingly, authenticated users may boot their computers without

waiting for an authenticated token to be present. Thus, authenticated users need not later wait through the bootup sequence if they neglect to place their authenticated token in the reader at startup.

[044] The OS-level check for an authenticated token allows an authenticated user to alternate between locked and unlocked modes at the OS level. The lock is activated simply by removing the token. For example, if a user temporarily leaves the computer and takes the authenticated token from the reader, the computer reverts to its locked mode. When the user returns and replaces the authenticated token in the reader, full functionality is restored.

[045] Now with reference to FIG. 6, the exemplary method for authenticating a user (as described above) may be integrated with a public key encryption scheme (*i.e.*, PKI) for encryption and/or decryption of data. Moreover, the token the user presents for authentication may be utilized in the PKI system for storing a user identity and public and/or private encryption keys. While the method of FIG. 6 will be described utilizing a Smart Card for the token, those skilled in the art will appreciate that any type of token, besides the specific example of a Smart Card, is contemplated by the principles of the present invention.

[046] Turning to FIG. 6, a Smart Card reader receives a Smart Card in step 600. The reader extracts a user identity from the Smart Card in step 602 and sends the user identity to third party certification authority 122 for validation in step 604. A determination is made whether the user identity is validated at third party certification authority 122 in step 606. Step 606 serves to verify the identity of the user presenting the Smart Card. If user identity is

not verified in step 606 ("No"), then the potential user is denied access to the PKI application, as shown in step 608. If the user identity is verified in step 606 ("Yes"), user is authorized to access the PKI application in step 610.

[047] As is known in the art, the user's public and private keys are periodically changed pursuant to the PKI process. A change interval for updating the public and private keys may be determined by a PKI administrator at third party certification authority 122. If a user is authorized to access the PKI encryption/decryption algorithm in step 610, and if the change interval has expired, updated public and/or private keys may be written to the Smart Card in step 612.

[048] FIG. 7 illustrates a flowchart of an exemplary method for checking to determine whether a user is authenticated to access files in static memory, such as the hard drive, consistent with an embodiment of the present invention. As mentioned previously, coding resident in the hard drive ensures that only authorized users have access privileges to certain files. Access privileges define the extent to which a user may access a file to, for example, view, modify, or create its contents. The access privileges for files on the hard drive are defaulted to a protected or inaccessible state. One skilled in the art may exclude some system files from this default inaccessible state so as to allow access to such files.

[049] At step 700, the computer receives a request to access a file from a user or from a program which the user is operating. In the exemplary embodiment, a software module, referred to as hard drive registry software, is then executed in step 702. As is known in the art, a "registry" is a repository for all the miscellaneous settings such as, for example, how computer

memory is set up and what application programs are to be present when the OS is started. "Registry software" refers to software capable of implementing those settings. Substantially coincidental with the execution of the hard drive registry software, an application executable call checks to determine whether an authenticated token is operatively engaged at token reader 114 (step 704). As a result of this application executable call, a determination is made whether an authenticated token is present at the token reader 114 in step 706. If the user is not authenticated to access the file, the access privileges are left unchanged, leaving the file inaccessible (step 708). If the user is authenticated to use the file, the access privileges for the file are changed from the default (*i.e.*, inaccessible) to an accessible state, as shown in step 710. Thus, the exemplary method of FIG. 7 prevents an unauthorized user from accessing the protected files.

[050] An aspect of the exemplary method of FIG. 7 is that the method may protect memory 104 even if it is physically removed from the computer. For example, if a hard drive is physically removed from a computer, an unauthorized user cannot install the hard drive on another computer and access the data thereon. Similarly, if a hard drive is physically removed from a computer, an unauthorized user cannot install the hard drive as a "slave" to another computer and access the data thereon. Thus, access to memory 104 is only available when an authenticated token is placed in token reader 116.

[051] Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. Those skilled in the art will appreciate that the principles of the present invention may be implemented in a variety of ways,

such as with hardware, software, or in a combination. Programs or resident code used to implement such embodiments of the present invention may be stored on computer-readable medium, such as RAM, ROM, removable hard drives, or any other memory storage device. Therefore, it is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims.

WHAT IS CLAIMED IS:

1. A method for protecting information stored on a computer having a bootup operation, an operating system, at least one input device, and at least one output device, comprising:

checking for a first authentication during the bootup operation of the computer, the first authentication being associated with an external device connected to the computer; and

locking the at least one input device of the computer if no first authentication is present during the bootup operation of the computer.

2. The method of claim 1, wherein checking for the first authentication further comprises checking for a first token.

3. The method of claim 2, wherein checking for the first token further comprises querying at least one of a Smart Card reader, a radio frequency (RF) tag reader, a magnetic stripe reader, a retinal scanner, a fingerprint reader, and a palmprint reader regarding the presence of the first authenticated token.

4. The method of claim 3, wherein the first token comprises at least one of a Smart Card, an RF tag, a token having a magnetic stripe, a retina from a user, a fingerprint from the user, and a palmprint from the user.

5. The method of claim 1, wherein checking for the first authentication further comprises checking for a first password.

6. The method of claim 1, further comprising:

checking for a second authentication when the operating system is operational; and

locking the at least one input device if no second authentication is present while the operating system is operational.

7. The method of claim 6, wherein checking for the second authentication further comprises checking for a second token.

8. The method of claim 7, wherein checking for the second token further comprises querying at least one of a Smart Card reader, a radio frequency (RF) tag reader, a magnetic stripe reader, a retinal scanner, a fingerprint reader, and a palmprint reader regarding the presence of the second token.

9. The method of claim 8, wherein the second token comprises at least one of a Smart Card, an RF tag, a token having a magnetic stripe, a retina from a user, a fingerprint from the user, and a palmprint from the user.

10. The method of claim 6, wherein checking for the second authentication further comprises checking for a second password.

11. The method of claim 9, wherein the first token and the second token are the same.

12. The method of claim 6, further comprising:

continuing the bootup operation even if the at least one input device is locked.

13. The method of claim 1, wherein the external device comprises a token reader.

14. The method of claim 1, further comprising locking the at least one output device.

15. The method of claim 6, further comprising locking the at least one output device.

16. A system for protecting information stored on a computer having a bootup operation and an operating system, comprising:

at least one input device;

an external device connected to the computer;

a checking module for checking for an authentication during the bootup operation of the computer, the authentication being associated with the external device; and

a locking module for locking the at least one input device of the computer if no authentication is present during the bootup operation of the computer.

17. The system of claim 16, wherein the checking module further comprises software instructions operational on the computer.

18. The system of claim 16, wherein the locking module comprises an instruction which locks a communication interface in the computer that connects the external device.

19. The system of claim 18, wherein the instruction engages a switch at the interface.

20. The method of claim 16, wherein the external device comprises at least one of a Smart Card reader, a radio frequency (RF) tag reader, a magnetic stripe reader, a retinal scanner, a fingerprint reader, and a palmprint reader regarding the presence of the authentication.

21. The system of claim 20, further comprising:

at least one output device, wherein the locking module locks the at least one output device of the computer if no authentication is present during the bootup operation of the computer.

22. A system for protecting information in a computer having a memory and at least one input device, the system comprising:

at least one token;

a reader for reading the at least one token; and

a first software operational in the memory for disabling the at least one input device of the computer during a bootup operation if the at least one token is not read by the reader.

23. The system of claim 22, further comprising:

a second software operational in the memory for disabling the at least one input device of the computer when an operating system is operational if the at least one token is not read by the reader.

24. The system of claim 22, further comprising:

a third software operational in the memory for disabling at least one output device of the computer when an operating system is operational if the at least one token is not read by the reader.

25. The system of claim 22, further comprising:

a fourth software operational in the memory for placing the memory in an inaccessible state if the authorized token is not read by the reader when the computer attempts to access the memory.

26. The system of claim 25, wherein the memory further comprises registry software; and wherein the fourth software further comprises an

application executable call executed substantially coincidentally with the registry software.

27. The system of claim 25, wherein the memory defaults to an inaccessible state.

28. The system of claim 27, wherein the memory changes to an accessible state if the authorized token is read by the reader.

29. A computer-readable medium operative in a computer system having a bootup operation, at least one input device, a token reader, and at least one authenticated token, the computer-readable medium containing a program for protecting information within the computer system by carrying out steps comprising:

checking for an authenticated token in the token reader during the bootup operation of the computer system; and

locking at least one input device if an authenticated token is not read by the token reader.

30. The computer-readable medium of claim 29, wherein the computer system comprises at least one output device; and

wherein the program further comprises locking the at least one output device if the authenticated token is not read by the token reader.

31. The computer-readable medium of claim 29, wherein the program further comprises:

checking for the authenticated token in the token reader when the operating system is operational; and

locking at least one input device if the authenticated token is not read by the token reader.

32. The computer-readable medium of claim 31, wherein the computer system further comprises at least one output device, and wherein the program further comprises locking the at least one output device if the authenticated token is not read by the token reader.

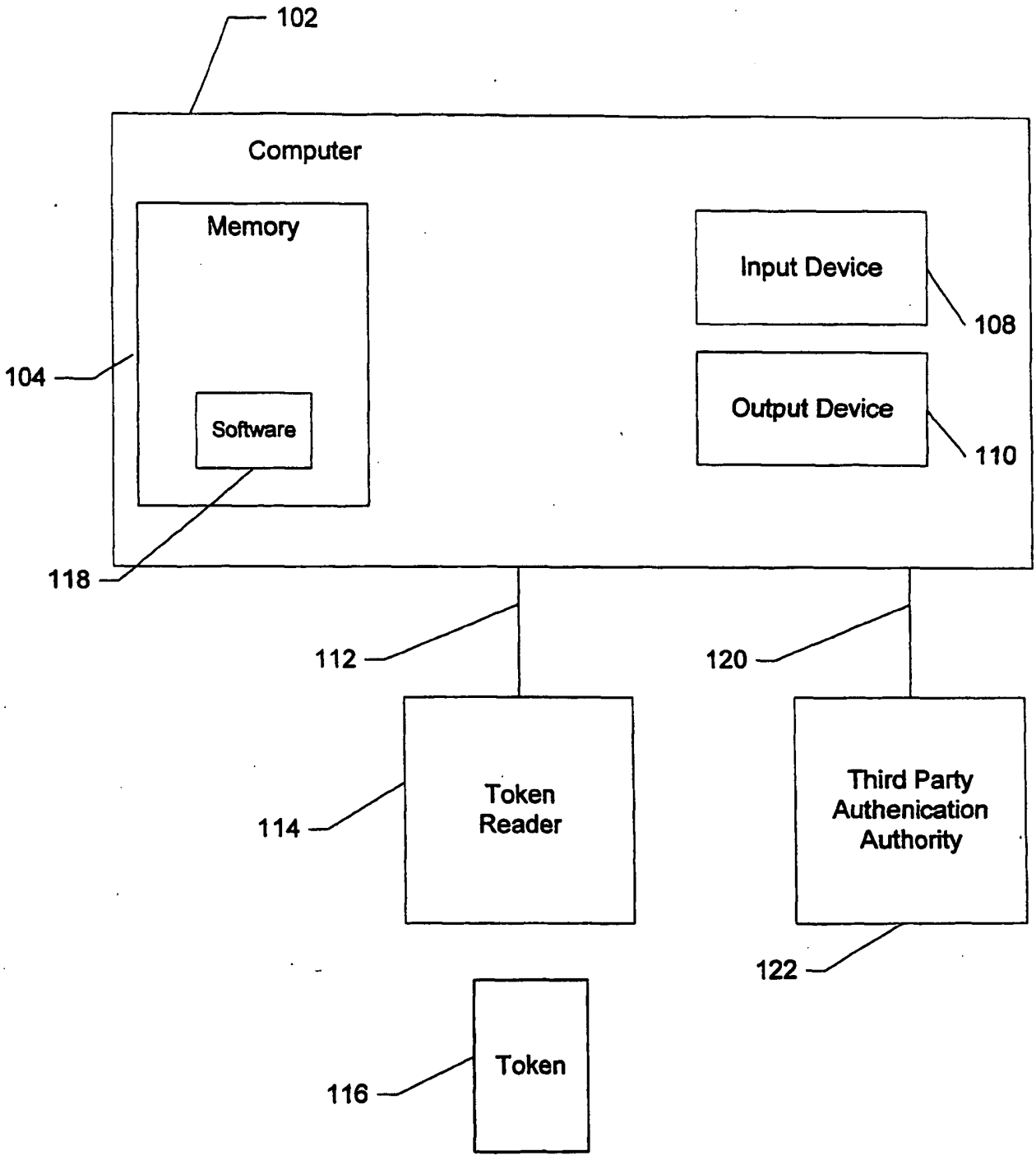
102
Computer
Memory
Software
104
118
Input Device
108
Output Device
110
112
120
Token Reader
114
Third Party Authenication Authority
122
Token
116

FIG. 1

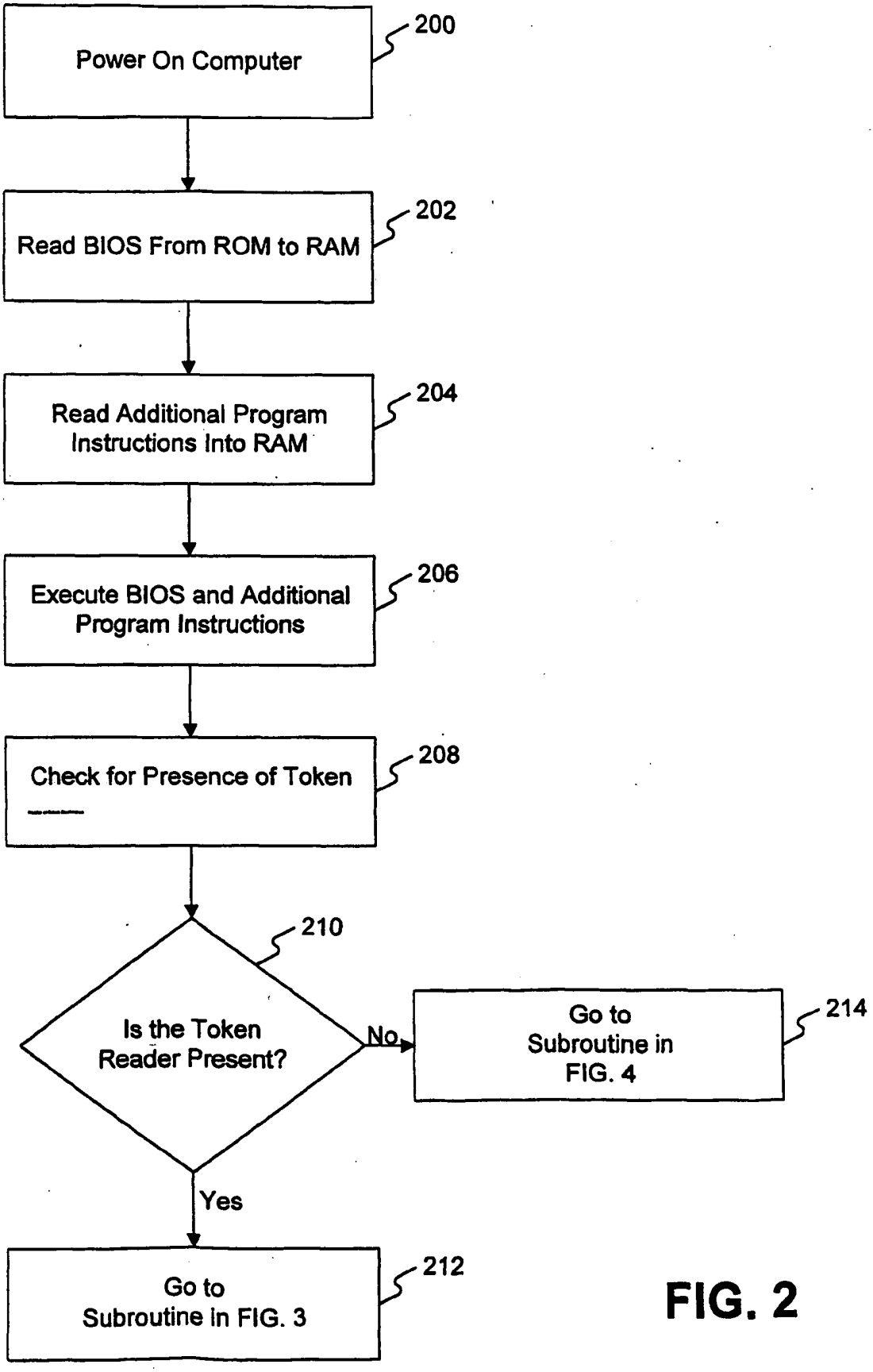
Power On Computer
200
Read BIOS From ROM to RAM
202
Read Additional Program Instructions Into RAM
204
Execute BIOS and Additional Program Instructions
206
Check for Presence of Token
208
Is the Token Reader Present?
210
No
Go to Subroutine in FIG. 4
214
Yes
Go to Subroutine in FIG. 3
212

FIG. 2

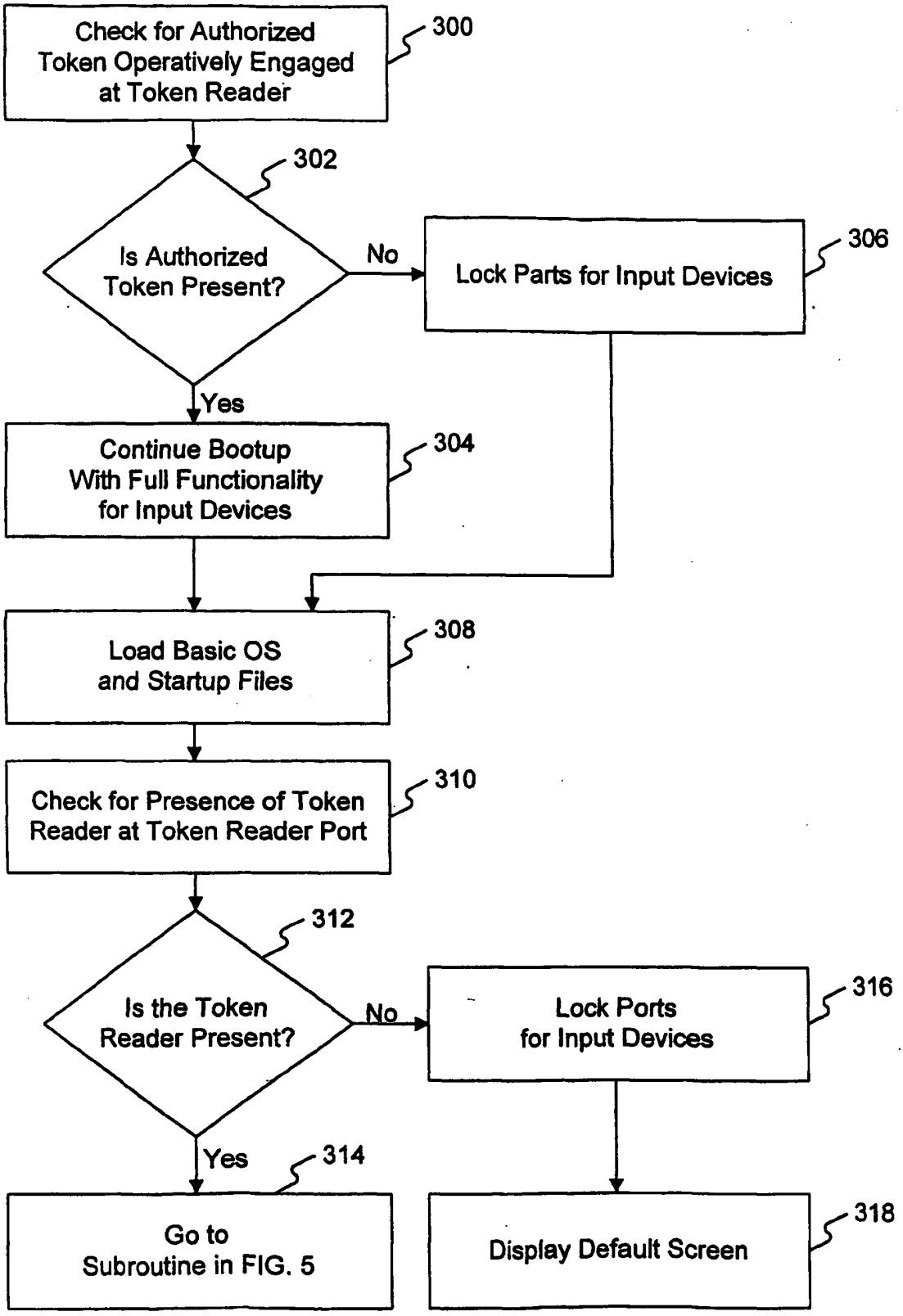
Check for Authorized Token Operatively Engaged at Token Reader
300
302
Is Authorized Token Present?
No
Lock Parts for Input Devices
306
Yes
Continue Bootup With Full Functionality for Input Devices
304
Load Basic OS and Startup Files
308
Check for Presence of Token Reader at Token Reader Port
310
312
Is the Token Reader Present?
No
Lock Ports for Input Devices
316
Yes
314
Go to Subroutine in FIG. 5
Display Default Screen
318

FIG. 3

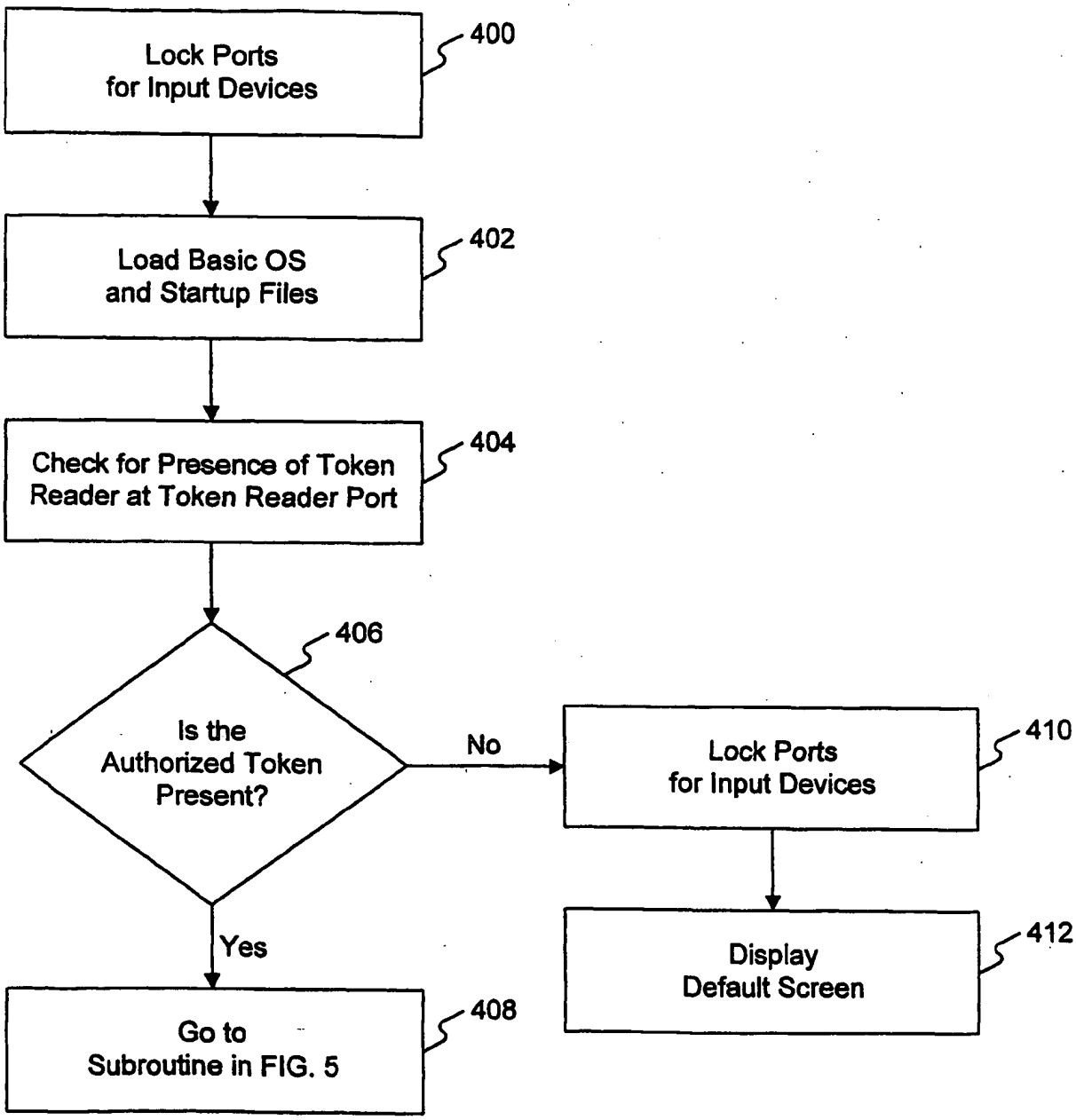
Lock Ports for Input Devices
400
Load Basic OS and Startup Files
402
Check for Presence of Token Reader at Token Reader Port
404
Is the Authorized Token Present?
406
No
Yes
Lock Ports for Input Devices
410
Display Default Screen
412
Go to Subroutine in FIG. 5
408

FIG. 4

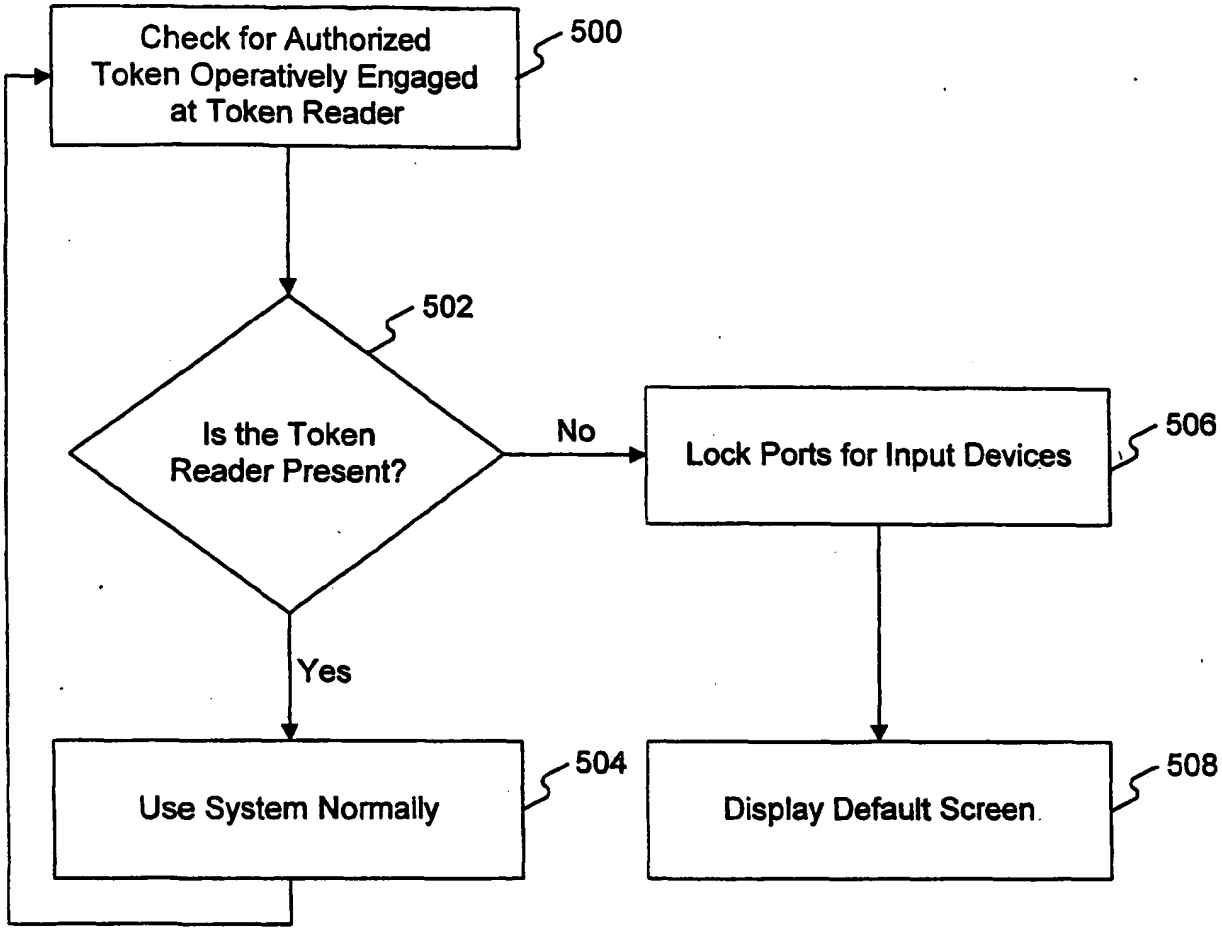
Check for Authorized Token Operatively Engaged at Token Reader
500
502
Is the Token Reader Present?
No
Lock Ports for Input Devices
506
Yes
Use System Normally
504
Display Default Screen
508

FIG. 5

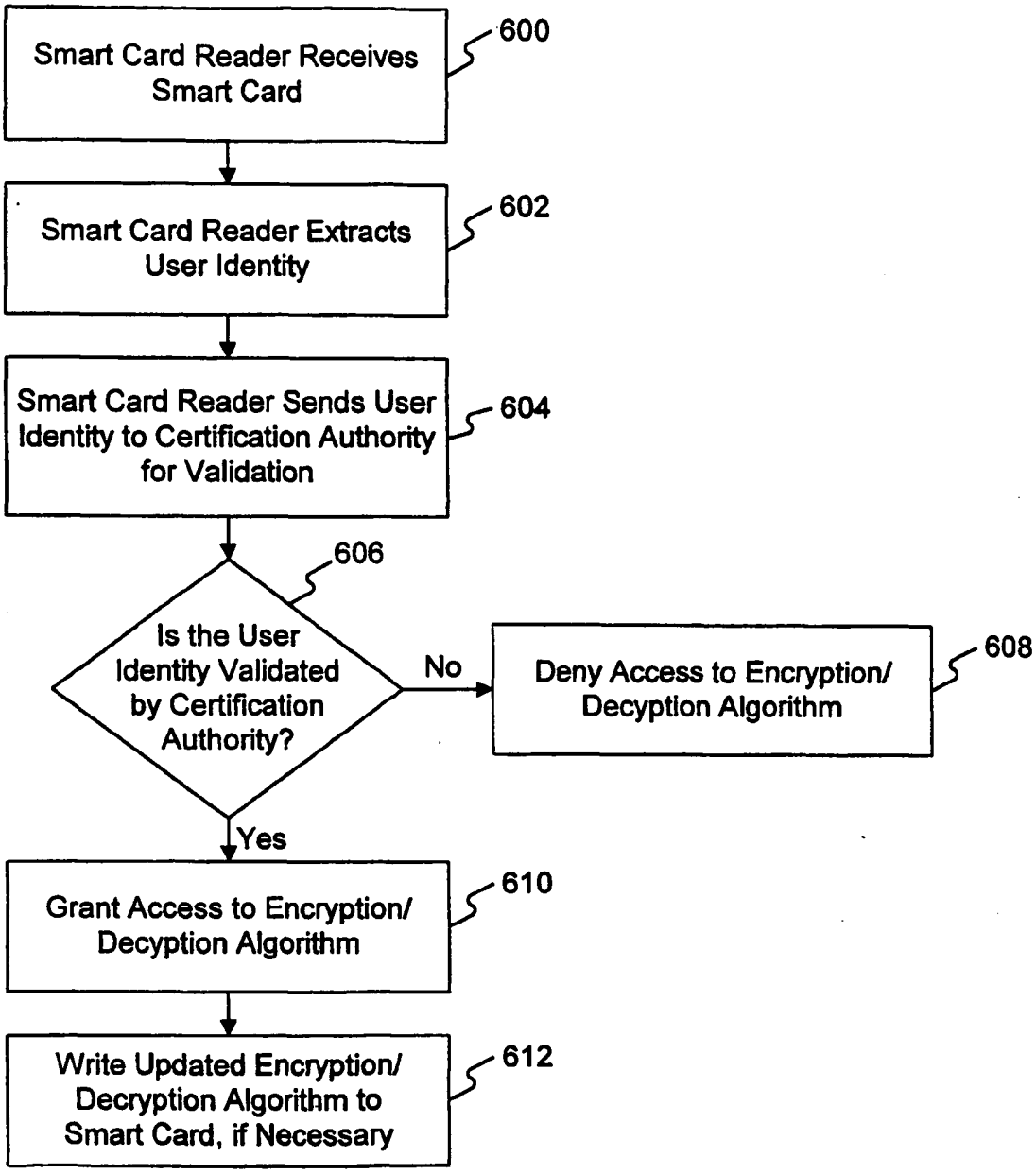
Smart Card Reader Receives Smart Card
600
Smart Card Reader Extracts User Identity
602
Smart Card Reader Sends User Identity to Certification Authority for Validation
604
606
Is the User Identity Validated by Certification Authority?
No
Deny Access to Encryption/ Decyption Algorithm
608
Yes
Grant Access to Encryption/ Decyption Algorithm
610
Write Updated Encryption/ Decryption Algorithm to Smart Card, if Necessary
612

FIG. 6

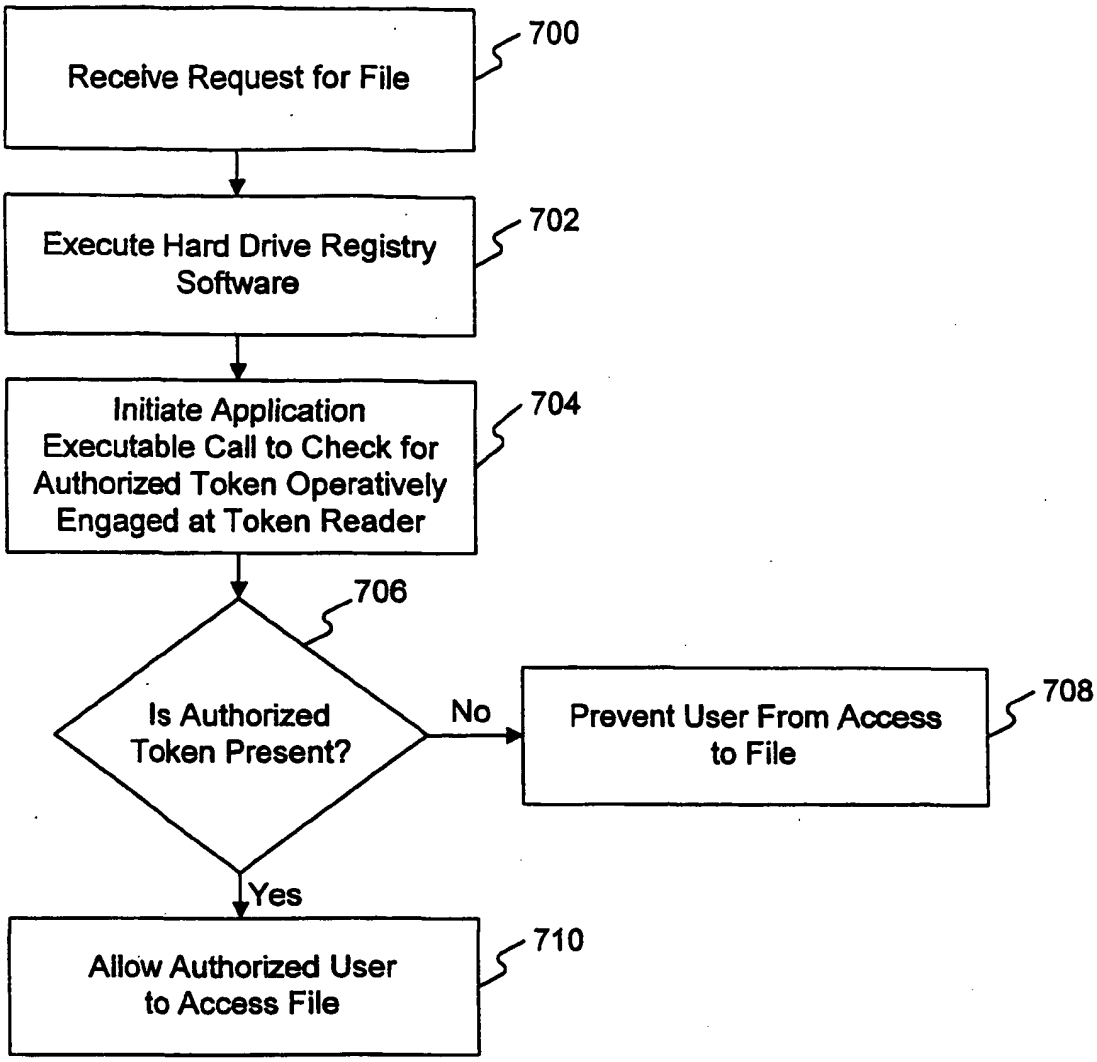
Receive Request for File
700
Execute Hard Drive Registry Software
702
Initiate Application Executable Call to Check for Authorized Token Operatively Engaged at Token Reader
704
706
Is Authorized Token Present?
No
Prevent User From Access to File
708
Yes
Allow Authorized User to Access File
710

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/28308

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :H04L 9/00
US CL : 380/278, 282, 285; 713/172, 173
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/278, 282, 285; 713/172, 173

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,949,882 A (Angelo) 07 September 1999, col. 6-8 and 13, lines 1-67.	1-32
A	US 5,375,243 A (Parzych et.) 20 DEC 1994, col. 1-10, lines 1-68.	1-32
A	US 5,748,888 A (Angelo et al.) 05 May 1998, col. 1-9, lines 1-67.	1-32
A	US 6,161,182 A (Nadooshan) 12 December 2000, col. 1-8, lines 1-67.	1-32
A	US 5,265,164 A (Matyas et al.) 23 November 1993, col. 1-46, lines 1-68.	1-32

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 12 NOVEMBER 2001	Date of mailing of the international search report 13 DEC 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer Albert DeCady James R. Matthews Telephone No. (703) 305-9595

Form PCT/ISA/210 (second sheet) (July 1998)*